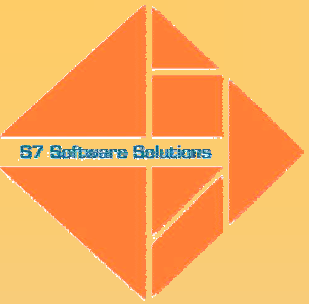


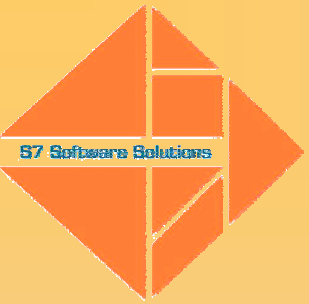
S7 Case Study:

Porting Fidelity's GT.M, A High-Performance Database Engine, to Itanium

Pankaj Kulkarni, Director of Engineering
pankajk@s7solutions.com

Steve Estes, Senior Architect, Fidelity National Information Services
Steven.Estes@fnis.com

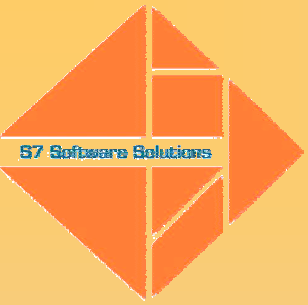
S7 Software Solutions
Re-Defining Cross-Platform Porting and Migration

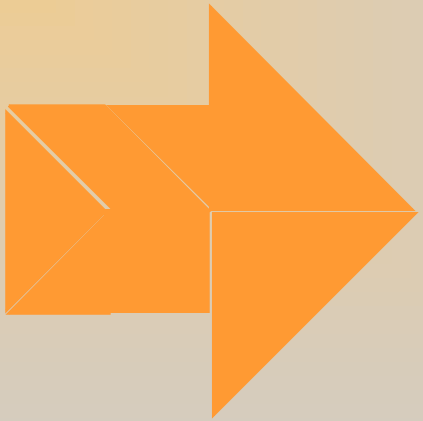


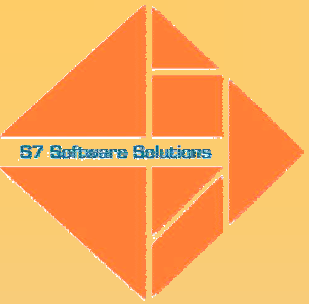
Agenda

- About S7
- About GT.M
- Challenges
- Technical Issues

About S7


S7 Software Solutions

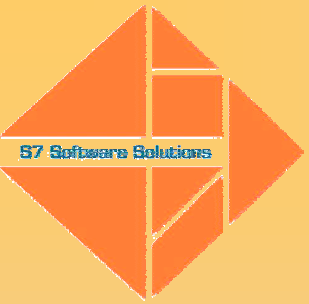




S7's Current Focus Areas

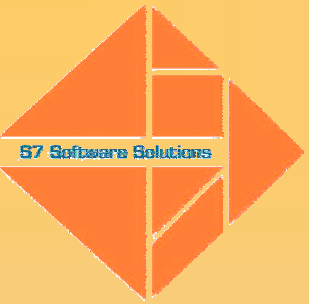
- Migration services in:
  - Itanium and 64-bit Migrations
  - Unix/Linux/Windows migrations
  - VB, VC++, OWL, VFP Migrations
  - Java/J2EE/.NET/C#/ASP Migrations
  - Motif/Qt/UI Migrations
- Migration tools development
- Multi-platform app development
- Multi-platform app testing

About GT.M



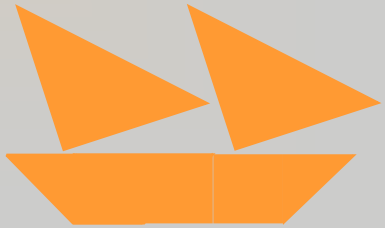
M/Mumps – what is it?

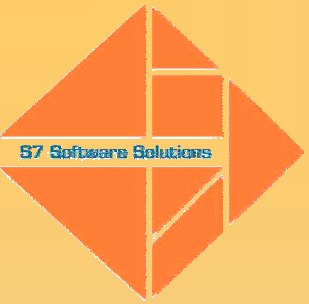
- Massachusetts General Hospital Utility Multi-Programming System
  - Common name: **MUMPS**
- ISO/IEC standard 11756:1999 – official name: M
- Widely used in healthcare
  - By virtually all major VARs – Epic, IDX (now part of GE), McKesson, Eclipsys...
  - By major institutions - Mayo, Kaiser, Cleveland Clinic, Partners, Quest, Lab Corp…
- Largest user is US Government
  - Dept. of Veterans Affairs, Dept. of Defense, Indian Health Service
- Banking and finance
  - Includes largest core processing system in the world
    – Krung Thai Bank
- Bottom line – a ubiquitous workhorse



M/MUMPS – standard features

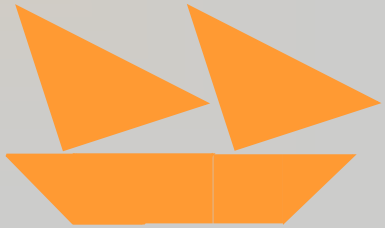
- Standard defines a language tightly bound to a data store
- Procedural Language
- Untyped variables, dynamically typed values
- Anything goes for subscripts and values
- A data store access is nothing more than an array access
- Features for real-world applications
  - e.g., robust error handling, dynamic execution
- Locks like traffic lights, not like doors



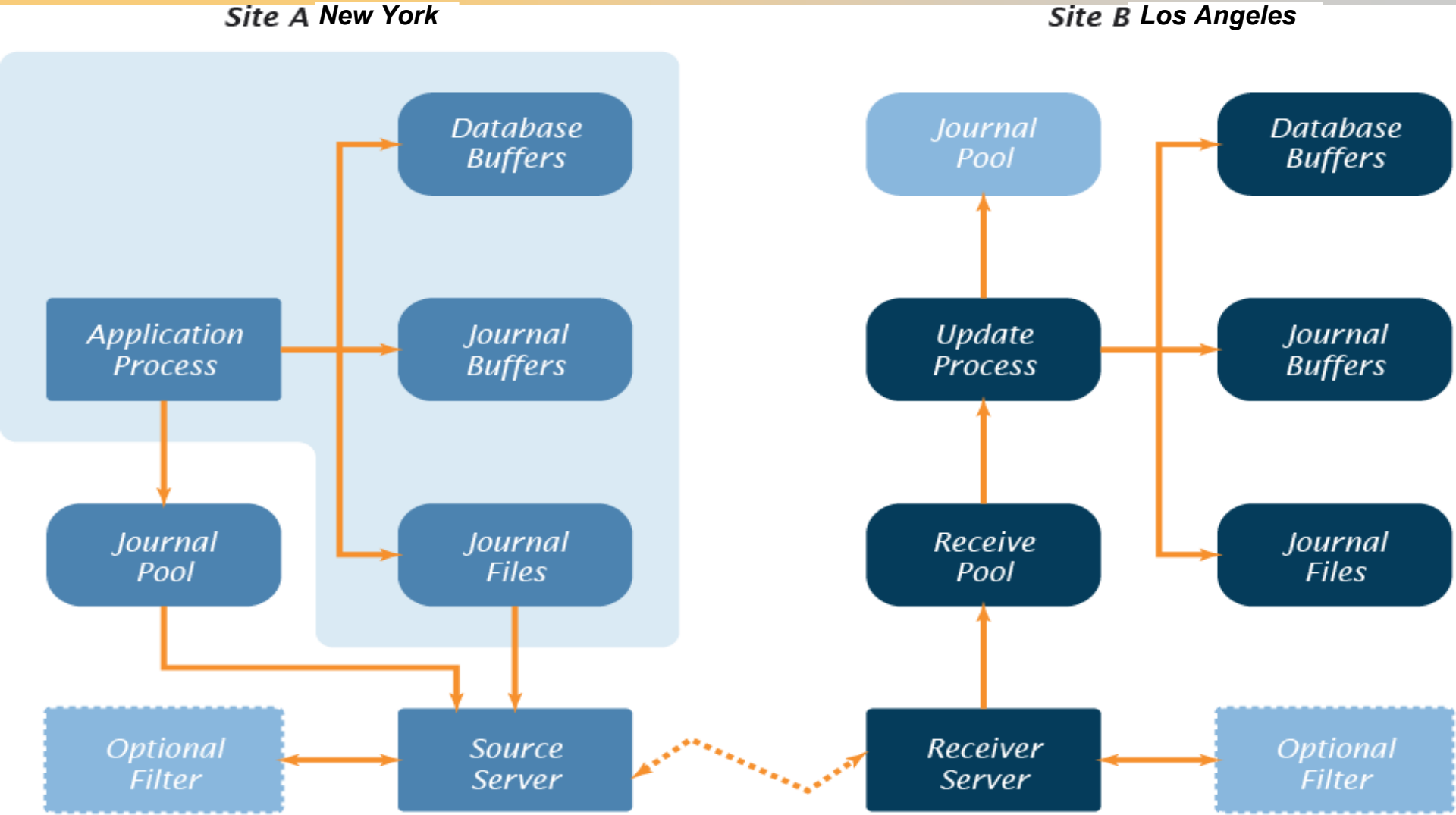


GT.M – more than MUMPS

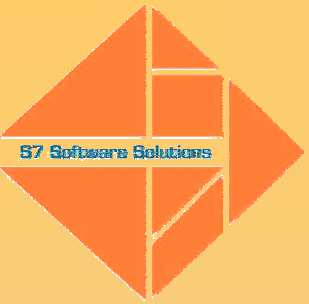
- Support for Unicode 5.0
- Daemon-less real-time database engine
  - Processes cooperate using run-time library
- Unique features for robustness and continuity of business
  - Not available on any other database engine
- Logical multi-site operation
  - 16 secondaries to root primary
  - each secondary can feed 16 tertiaries
  - and so on...



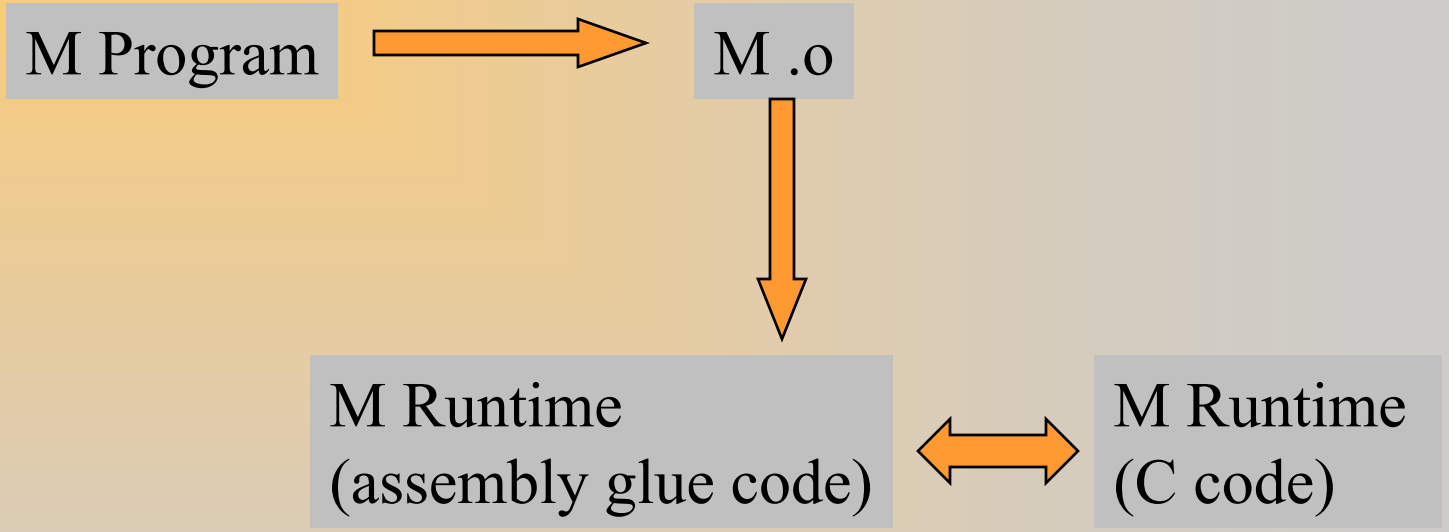
Logical dual site

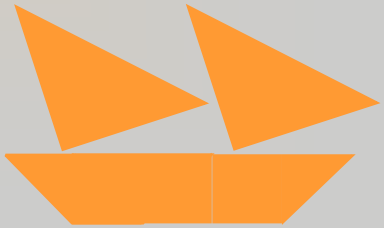


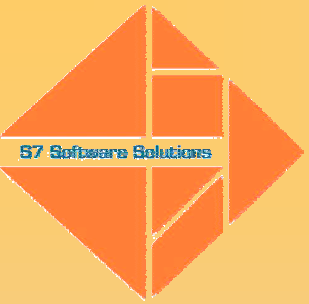
Challenges



GT.M Runtime Architecture

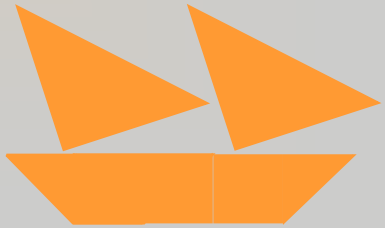


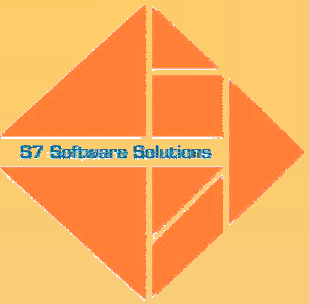




Project Requirements

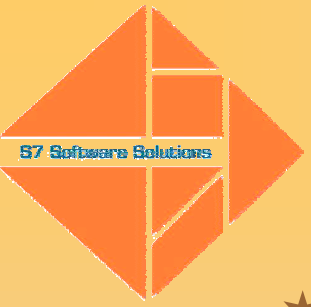
- Own runtime architecture. Different from Itanium ABI.
- Run time generation of code.
- Write instructions into data segment and execute.
- Efficient locking
- Should work both on HPUX and Linux
- 64-bit enablement
- Minimal disruption to existing platforms
- Assembly file conversion
- aCC on HPUX, gcc on Linux





Strategy

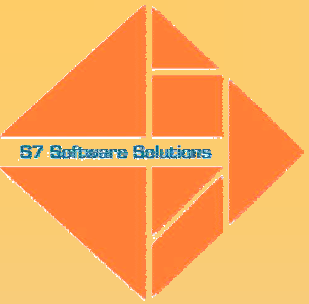
- Iterative process.
- Compiler/Linker/Loader first.
- Database portions later.
- Smoketest
- GDE (written in M)
- Test Suite



Effort

- 5 Senior developers, 8 months
- Parallel development on HPUX and LINUX
- Incremental effort to port to Linux – 15%

Technical Issues



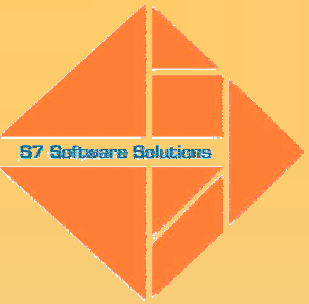
Assembler differences

- gcc vs aCC
  - br.call
  - #include
  - Expansion of ##

    #define foo(a,x) a##x

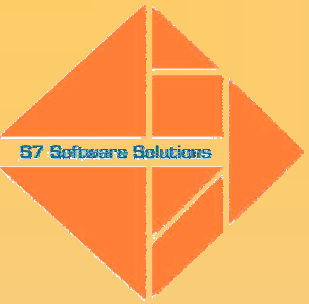
    foo(i, j) ✗

    foo(i,j) → Correct
- Used pseudo macro/operations in assembly files
  - to improve the portability
  - reusability



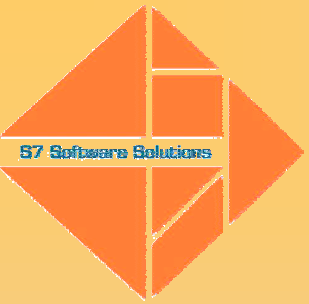
Code generation issues

- Instructions always encoded in little endian format
  - irrespective of the endianess mode currently active)
- Decoding the instructions is tricky
  - 128 bit instruction bundles
    - each instruction is of size 41 bits
    - 5 bits are used by a template identifier



Code generation issues.. (Contd)

- Instructions can be larger than 41 bits
  - eg: movl –move a 64 bit immediate value to a register.
- Decoding instructions requires
  - bit-field structures,
  - 3 instructions at a time.
- Many instructions share the same opcode
  - Need to use other extension fields of the instruction.



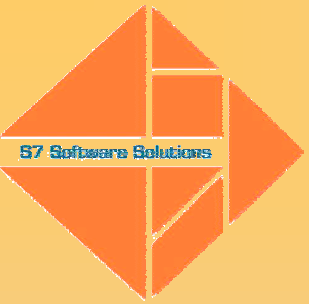
Code generation.. (Contd)

- Many compare instructions ➔ a single instruction
  - by just interchanging the operands
  - Using the predicate bits

if (a < b) goto X
cmp.lt p1, p2 = a,b
(p1) br X

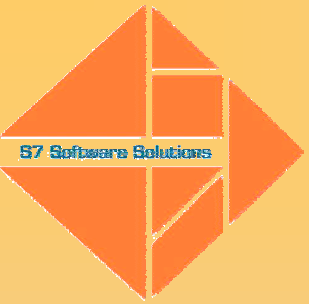
if (a > b) goto X → **if (b < a) goto X**
cmp.lt p1,p2 = b,a
(p1) br X

if (a >= b) goto X → **if (!(a<b)) goto X**
cmp.lt p1,p2 = a,b
(p2) br X



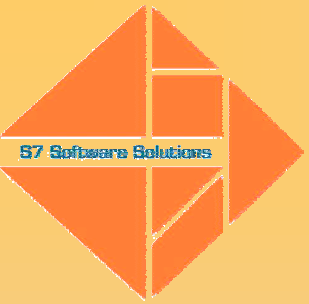
Code generation... (contd)

- App works only in debugger with stepi. Why?
  - Stop bit set incorrectly in the template for the bundle.
  - Data dependency



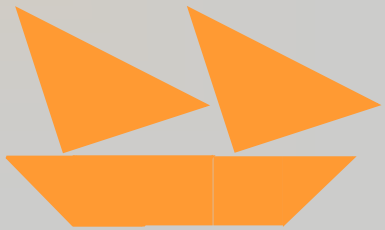
Porting assembly files

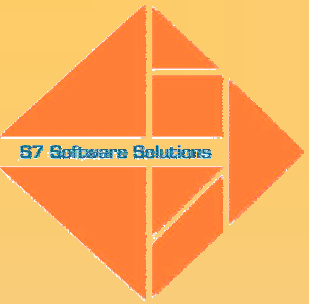
- The bit positions are different in PA and IA.
  - On PA, bit 0 represents the MSB.
  - On IA64, bit 0 represents the LSB.
- No delay slot in IA64.



Porting Assembly files... (contd)

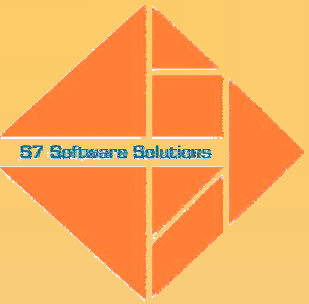
- Difference in calling a C function vs calling an assembly function.
  - On IA64, a function call is made using br.call.
  - br.call also modifies the RSE (register stack engine).
  - A function called using br.call, should always return back using br.ret
  - C compiler generates a br.ret when returning back.





Porting Assembly files… (contd)

- Assembly routines might not always return directly back to the caller.
- br.call when calling a C function from assembly
- ‘br’ when calling assembly functions
- Porting from PA to IA is easier than porting from IA to PA
  - PA does not have the RSE concept

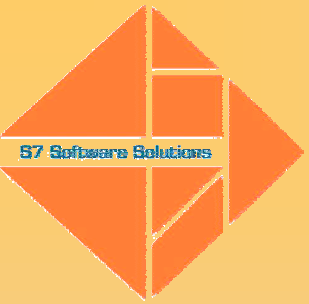


Endianness Issues

- Linux is little endian, HPUX is big endian
- Byte swapping when constructing the instruction (on HP)

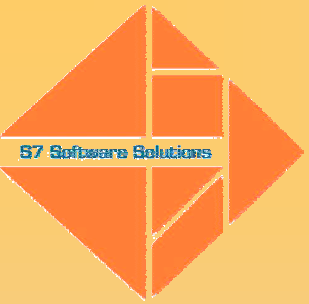
```
#define GTM_BYTESWAP_64(LL)                          \
        (  (((LL) & 0xff00000000000000ull) >> 56)     \
         | (((LL) & 0x00ff000000000000ull) >> 40)     \
         | (((LL) & 0x0000ff0000000000ull) >> 24)     \
         | (((LL) & 0x000000ff00000000ull) >>  8)     \
         | (((LL) & 0x00000000ff000000ull) <<  8)     \
         | (((LL) & 0x0000000000ff0000ull) << 24)     \
         | (((LL) & 0x000000000000ff00ull) << 40)     \
         | (((LL) & 0x00000000000000ffull) << 56)     \
        )
```

- Reordering of bit fields (esp in the instruction templates) was required.



Endianness Issues.. (Contd)

```
typedef union
{
        struct {
                uint8 aValue;
                uint8 bValue;
        }hexValue;
        struct {
#ifdef BIGENDIAN
        uint8       inst2lo:18;
        uint8       inst1:41;
        uint8       template:5;
        uint8       inst3:41;
        uint8       inst2hi:23;
#else
        uint8       template:5;
        uint8       inst1:41;
        uint8       inst2lo:18;
        uint8       inst2hi:23;
        uint8       inst3:41;
#endif
        }format;
} ia64_bundle;
```



Pointer truncation (64-bit)

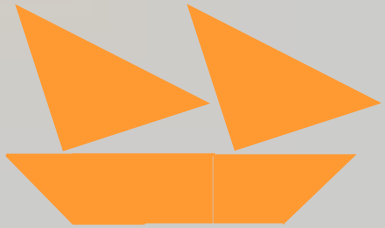
- Address stored in integer variables

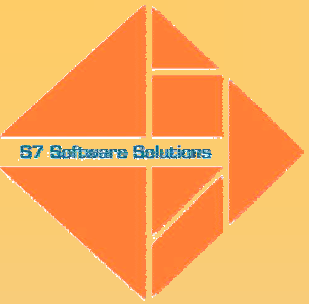
Eg:

```
int addr = base + offset;
```

- Use intptr_t and uintptr_t (from /usr/include/inttypes.h) for variables that store addresses, to ensure portability

- +M2 option of HP C Compiler





C-advise

- C-advise reports a variety of 64 bit migration warnings.

- Strict even under explicit casting.

Eg:

long b;

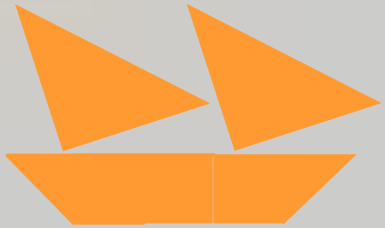
b = (long) -1; → *This will trigger a C-advise migration warning, saying a 32 bit integer is being assigned to a 64 bit value.*

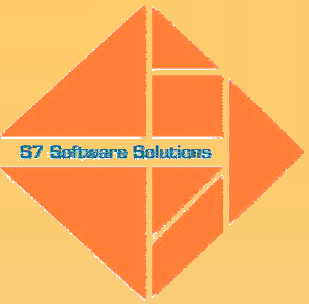
Solution:

b = (long)-1L;

Alignment issues

- Instruction address must start on 16 byte boundary
- Structure elements aligned for performance reasons
- Malloc returns 16 byte aligned address.
  - M_MXFAST, M_GRAIN



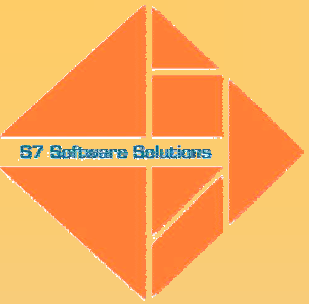


Differences between HP-UX and Linux

- gcc does not sign extend the return register, for integer returns, by default

```
int4 numcmp(int x, int y)
{
        if (x < y) return -1;
        if (x==y) return 0;
        if (x > y) return 1;
 }
 op_numcmp.s:

 mov r44 = 0
 mov r45 = 1
 br.call.sptk.many b0 = numcmp
 cmp.lt  r8, r0   <<<<----- What will be the value of r8 ???
 br lesser;
```

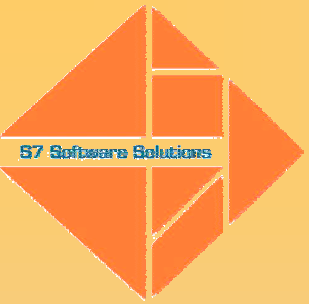


Differences between HP-UX and Linux.. (Contd)

- Writing executable code in malloced regions.
  - On HP, by default malloc regions have execute permission set.
  - On Linux, due to security concerns, malloc regions do not have execute permission turned by default.
    - Use mmap directly, and explicitly specify the permissions needed.
    - Or use memalign and mprotect, to have executable regions in the heap (malloc) region.

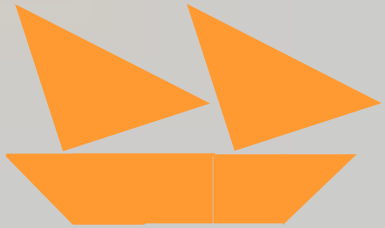
Eg:

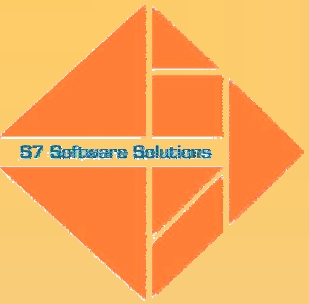
```
char *ptr = memalign(sysconf(SC_PAGESIZE), size);
mprotect(ptr, size, PROT_READ | PROT_WRITE | PROT_EXEC);
```



Tools used (from HP)

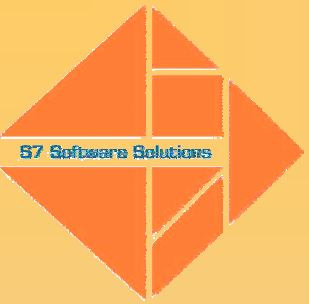
- Software Transition Toolkit
  - www.hp.com/go/STK
- CAdvise (+w64bit option)
  - www.hp.com/go/aCC
- Caliper
  - www.hp.com/go/caliper





Summary

- Many low level details of the Itanium architecture, while very powerful, can be difficult to debug.
- Incremental porting effort for HP-UX/Linux is around 10-15%.
- Following Iterative process during construction phase is ideal.
- Having a ready test-suite on source platform helps a lot.



MUMPS, GT.M References

- http://www.fis-gtm.com
- http://sourceforge.net/projects/sanchez-gtm
- http://en.wikipedia.org/wiki/MUMPS
- http://www.faqs.org/faqs/m-technology-faq/
- http://worldvista.org
- http://sourceforge.net/projects/worldvista
- http://207.192.157.194/demo/TestCGI.htm
- http://vista.vmth.ucdavis.edu/home/index/48.html
- http://mjsp.sourceforge.net
- http://mgateway.com/php/mgw/ewd.php
- Wikipedia – http://en.wikipedia.org/wiki/MUMPS
- FAQ – http://www.faqs.org/faqs/m-technology-faq/

For GT.M, K.S. Bhaskar contact information

ks.bhaskar@fnis.com

+1 (610) 578-4265

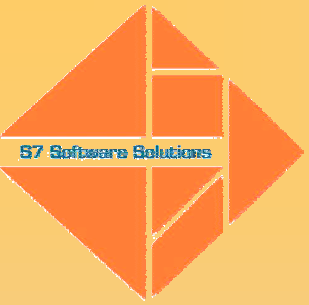
Contact Us

#9, 3rd Floor, 100 Ft Rind Road,
27th Main, BTM Phase I,
Bangalore – 560068, India
Tel : +91 80 41526777
Fax: +91 80 23341345

2936 173rd Ct NE,
Redmond, WA 98052,
USA
Tel : (425) 867 1457
Fax: (425) 883 2597

UK : (845) 0092 364
US Toll Free: (888) 224 6175

www.s7solutions.com
info@s7solutions.com



Thank You!!

… Questions?

Backup Slides

S7 FOCUS

Your Application

S7 PORT

Microsoft .net

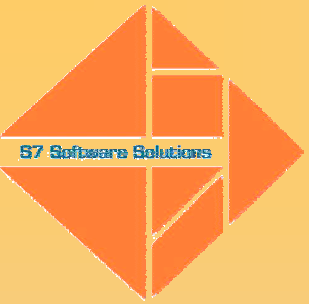
JAVA

redhat

suse

intel inside ITANIUM 2

AMD 64 Opteron



Atomic locking

- Atomic instructions are available for implementing lock/unlock functionality.

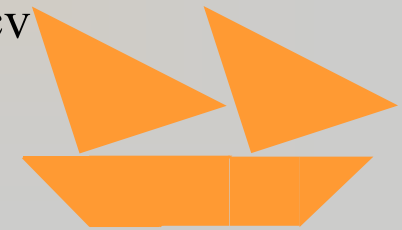
Eg:

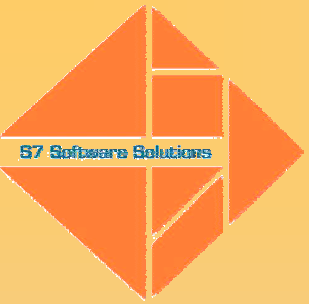
mov ar.ccv = REG_ARG1;;

cmpxchg4.**acq**.nta REG_RET0=[REG_ARG0], REG_ARG2, ar.ccv

Hints can be used to specify whether the instruction is used for locking or unlocking operation

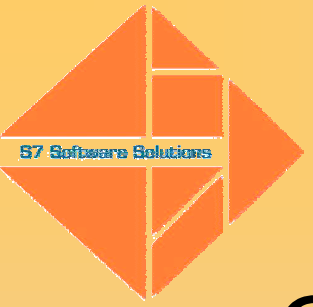
cmpxchg4.**rel**.nta REG_RET0=[REG_ARG0], REG_ARG2, ar.ccv





LP32 and LP64 size differences

C data type	ILP32	LP64
Char	8	Unchanged
Short	16	Unchanged
Int	32	Unchanged
Long	32	64
Long long	64	Unchanged
Pointer	32	64
Enum	32	Unchanged
Float	32	Unchanged
Double	64	Unchanged
Long double	128	Unchanged



Tools

CVS

Bugzilla – Bug Tracking and Artifacts.

Email, Phone

Status updates and interaction
Between HP, Fidelity and S7.